Predictions from Random Effects Models

Instructions

Today's guided practice exercise will lead you through fitting, assessing and interpreting a mixed-effect model.

Before you start, elect in your group:

- An R operator/code typist/screen sharer
- A manager to keep everyone on task and on schedule
- A communication person to take notes, talk with prof as needed
- If you have more people: a course notes/slides/Moodle reference person

Then, have fun!

As you work, write your answers to questions (and you comments about model selection graphs, etc.) in your Quarto doc. You will hand in your rendered version at the end of in-class work time for assignment credit.

Data

Read in the data (you don't need to interpret/understand the tidying being done here):

```
pw <- read_csv('https://sldr.netlify.app/data/pwd.csv') |>
  mutate(activity = ifelse(n_buzzes > 0,
                           'foraging',
                           'other'),
         activity = forcats::fct_relevel(activity,
                                         'other',
                                         'foraging'))
pw <- pw |>
  mutate(prev_buzzes = c(0, head(pull(pw, n_buzzes), -1)),
         time_cat = cut_interval(time_of_day, length = 0.1))
```

The research question you will consider is: What characteristics of pilot whale dives help you to predict whether they are foraging (hunting for food) or doing something else?

Use response variable `activity`.

The data you have were collected with animal-borne electronic tags, and the dataset has 6452 rows. Each row is one dive by an individual whale, and there are dives from 14 whales in the dataset. (So - comfortably similar to the beaked whale data you saw in class in many ways…)

Possible predictors include:

- `ODBA` (Overall Dynamic Body Acceleration - a measure of how much energy the animal is using to move and swim),
- dive `duration` in seconds,
- dive `depth` in meters,
- sonar `exposure` (noise nearby that might affect how whales behave),
- `dive_state` (behavior type, as judged by biologists watching whales when they are at the sea surface),
- whale's size (`ind_size`),
- `time_of_day` in decimal hours,
- `time_cat` (time as categorical variable, in one-hour blocks – *do not* use both time of day variables in same model!),
- `group_size` (number of other whales in same group),
- `water_depth` at the whale's location,
- `prev_buzzes` (number of suspected feeding events in the *previous* dive),
- `individual` (a unique identifier for each whale that was studied).
- (**Ignore** `sound_level` and `n_buzzes` – do not use those variables.)

Model Fitting

For this exercise, to save time, you will not plan the model yourself.

Instead, consider this model:

```r
pw_re <- glmmTMB(activity ~ ODBA + duration +
                   prev_buzzes*dive_state +
                   (1 | individual/time_cat),
                 data = pw, family = binomial(link = 'logit'))
summary(pw_re)
```

```
 Family: binomial  ( logit )
Formula:          activity ~ ODBA + duration + prev_buzzes * dive_state + (1 |
    individual/time_cat)
Data: pw

     AIC      BIC   logLik deviance df.resid
  2493.4   2601.8  -1230.7   2461.4     6436

Random effects:

Conditional model:
 Groups              Name        Variance Std.Dev.
 time_cat:individual (Intercept) 4.2108   2.0520
 individual          (Intercept) 0.4257   0.6524
Number of obs: 6452, groups:  time_cat:individual, 653; individual, 14

Conditional model:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -5.412e+00  5.737e-01  -9.434  < 2e-16 ***
ODBA                           -8.855e-03  7.786e-02  -0.114   0.9095
duration                        1.951e-02  2.759e-03   7.071 1.54e-12 ***
prev_buzzes                     9.433e-01  3.906e-01   2.415   0.0157 *
dive_stateDirected              4.248e-01  4.925e-01   0.863   0.3883
dive_stateForaging             -9.882e-01  1.030e+00  -0.960   0.3372
dive_stateScouting              9.794e-01  4.463e-01   2.194   0.0282 *
dive_stateSurf                  7.216e-01  4.497e-01   1.605   0.1085
dive_stateTravel               -1.450e-02  4.639e-01  -0.031   0.9751
prev_buzzes:dive_stateDirected -6.655e-02  4.830e-01  -0.138   0.8904
prev_buzzes:dive_stateForaging  1.774e+01  3.268e+03   0.005   0.9957
prev_buzzes:dive_stateScouting -7.911e-01  4.404e-01  -1.796   0.0725 .
prev_buzzes:dive_stateSurf     -7.845e-01  3.937e-01  -1.993   0.0463 *
prev_buzzes:dive_stateTravel   -7.070e-01  3.991e-01  -1.772   0.0765 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Discuss with your group:

- What **random effects** and what **fixed effects** are included in this model?
- Are the random effects nested, or not?
- **How many coefficients and parameters have to be estimated to fit the model above? Don't forget the variances (for the random effect(s) and the residuals)...** *Hint: use the model `summary()` if needed to help count! And for a binary*

regression, there is no parameter being estimated for the residual variance or dispersion (no equivalent of a `lm()`'s $\sigma_{residuals}$)

- **Now, what if you had included `individual` as a fixed effect instead? With 14 whales in the dataset, how many more coefficients would it take to add the `individual` predictor as a fixed effect instead of a random effect?**

Consult Prof DR about your answers. Make sure everyone understands how you determined the answer.

Model Planning check-in

According to our previous rule of thumb, how many coefficients could we safely plan to estimate from this data set? To figure it out, create a table showing the number of "successes" and "failures" in the dataset.

```
mosaic::tally(~activity, data = pw)
```

```
activity
   other foraging
    5930      522
```

What do you think - are we estimating too many coefficients, or are we OK?

When fitting random effects models, *be conservative* about the number of coefficients you're including. Fitting a random effect technically estimates just one parameter value, the random effect variance σ_{RE}. But many argue that it "uses up" an amount of coefficient-fitting capacity that is somewhere between 1 (for the random effect variance) and the number of coefficients it would take to include that same predictor as a regular predictor (here, 13).

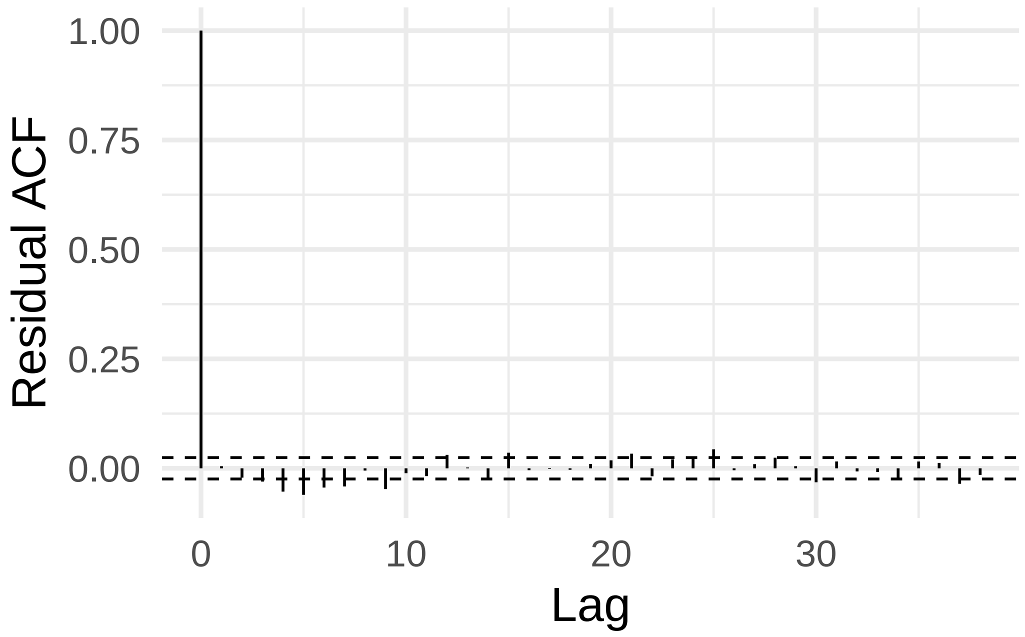
So when fitting random-effects models, we might err a bit on the side of including fewer predictors.

Model Assessment

Code to produce graphs is provided in this section (again, to save time). Your job is to interpret the graphs.

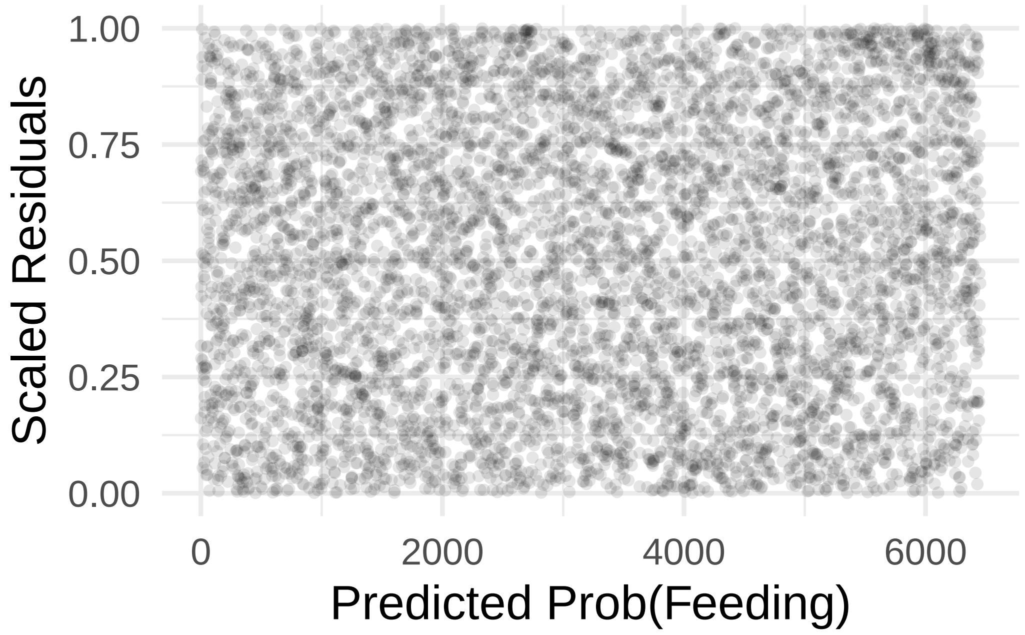
Independence of residuals

```
gf_acf(~pw_re)
```



Mean-variance relationship and (logit) linearity

```
pw_sim <- simulateResiduals(pw_re)
gf_point(pw_sim$scaledResiduals ~
            # if you are going to "plot your own" scaled residuals
           # for a model with random effects, get fitted values
           # EXCLUDING REs with input re.form = NA.
           # the rank() (as shown in count model notes) scales the x axis :
           rank(predict(pw_re, re.form = NA, type = 'response')),
         # make points semi-transparent
         alpha = 0.1) |>
  gf_labs(x = 'Predicted Prob(Feeding)',
          y = 'Scaled Residuals')
```

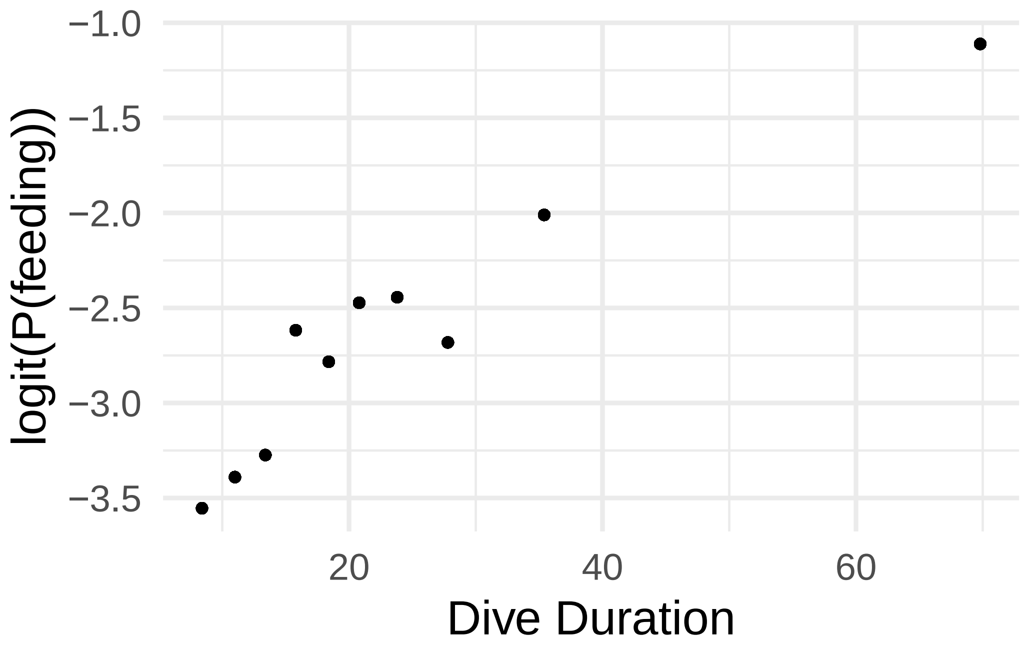


(logit) Linearity

In class we mentioned that for one-trial-per-row binary data, it's a bit beyond the scope of our class to do the data prep needed to make a plot of the *data* to check logit linearity.

But for fun and for your consideration, code is given here that shows one example of checking linearity on a per-predictor basis:

```
pw_dur <- pw |>
  mutate(cat_duration = cut_number(duration, 10)) |>
  group_by(cat_duration) |>
  mutate(prop_feeding = prop(~activity == 'foraging'),
         median_dur = median(~duration),
         logit_p_feeding = logit(prop_feeding))

gf_point(logit_p_feeding ~ median_dur,
         data = pw_dur) |>
  gf_labs(x = 'Dive Duration',
          y = 'logit(P(feeding))')
```



(If you have time at the end, try this for one more predictor).

Looking at all the plots above, what do you think – any evidence that conditions are not met?

Consult with Prof DR about your group's conclusions before proceeding further.

Prediction Plots: Average RE Group

First, we make the predictions that are most natural and simple to make: ones for "the average random effect group". These are equivalent to using the right-hand side of the model equation (*ignoring* both ϵ_{RE} and ϵ_{resid}) and applying the inverse logit transform to convert it to a probability.

Since we have two random effects: time period nested within individual, here we'll be making predictions for the *average* time-period of the *average whale*.

We have to choose one predictor to make the first predictions plots for; I'll choose dive `duration`.

```
library(ggeffects)
ggpredict(pw_re,
          terms = 'duration')
```

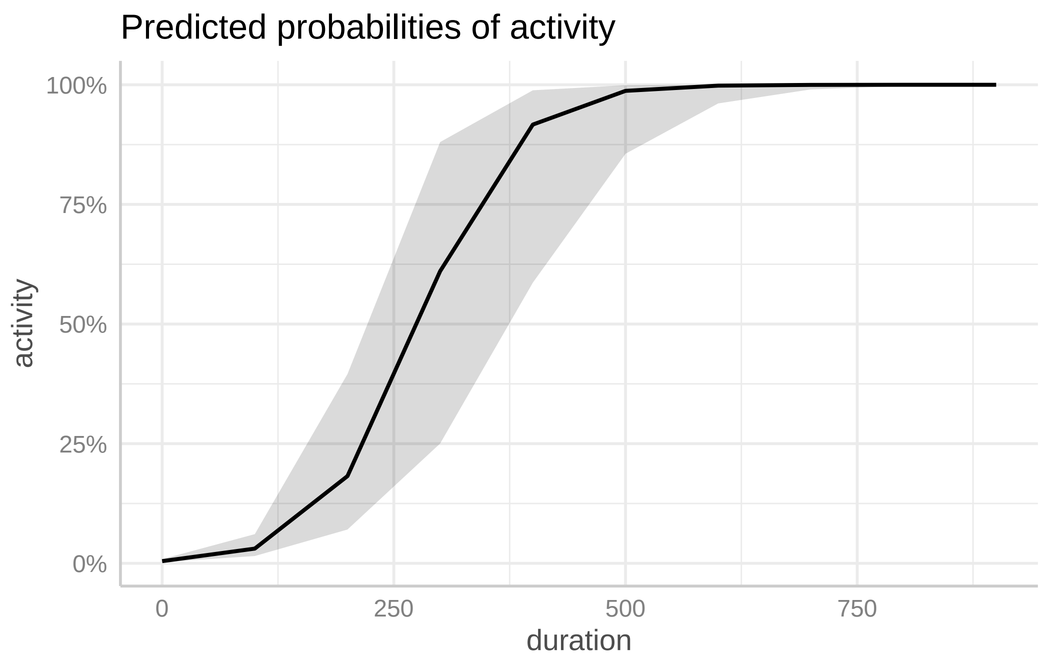
```
Data were 'prettified'. Consider using `terms="duration [all]"` to get
  smooth plots.

# Predicted probabilities of activity

duration | Predicted |     95% CI
---------------------------------
       0 |      0.00 | 0.00, 0.01
     100 |      0.03 | 0.02, 0.06
     200 |      0.18 | 0.07, 0.40
     300 |      0.61 | 0.25, 0.88
     500 |      0.99 | 0.86, 1.00
     600 |      1.00 | 0.96, 1.00
     700 |      1.00 | 0.99, 1.00
     900 |      1.00 | 1.00, 1.00

Adjusted for:
*        ODBA =   2.05
* prev_buzzes =   0.18
*  dive_state = Travel
*    time_cat = NA (population-level)
*  individual = NA (population-level)


Not all rows are shown in the output. Use `print(..., n = Inf)` to show
  all rows.
```

At what values were the "predictors not shown" fixed? They are in the output above, under the heading "Adjusted for".

To get a graph rather than text output (now that we know the fixed values used to make the predictions), just add `|> plot()` to the code.

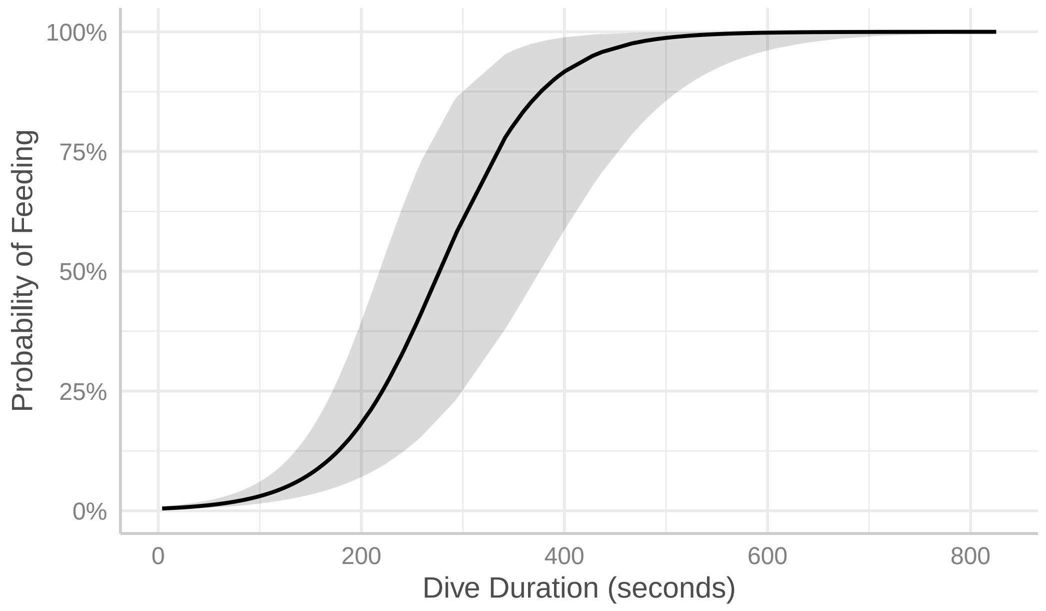
```
ggpredict(pw_re,
          terms = 'duration') |>
  plot()
```

```
Data were 'prettified'. Consider using `terms="duration [all]"` to get
  smooth plots.
```



Taking `ggeffects` up on its offer to get a “smooth plot” which will be prettier:

```
ggpredict(pw_re,
          terms = 'duration [all]') |>
  plot() |>
  gf_labs(title = '',
          y = 'Probability of Feeding',
          x = 'Dive Duration (seconds)')
```



Summarize the relationship that exists between dive duration and probability of feeding, according to this model, in a sentence.

Prediction Plots: All the Whales

This isn't something you need to be able to do or that you will do for a test or project!

But I think it may help with understanding the idea of the random effect, and also the "average individual".

So: I'll show you how one could make predictions for *each* of the whales in the dataset, and we'll add them to our "average RE group" prediction plot.

First, we make a hypothetical dataset where duration varies from 4 to 825 seconds (like in the real data), for each of our 14 whales. We'll fix the other predictors at the same values as before.

```
ind_pred_data <- expand.grid(individual = pull(pw, individual) |> unique(),
                             duration = seq(from = 4, by = 10, to = 825),
                             ODBA = 2.05,
                             water_depth = 450.57,
                             prev_buzzes =   0.18,
                             dive_state = 'Travel',
                             time_cat = NA) #(population-level)
```

Now, make predictions with SEs for this dataset, but *including* the random effect of individual. (R is able to estimate, for each individual, how far it is from the "average" individual; the standard deviation of these differences is the reported estimate of σ_{RE}!)

```
ind_preds <- predict(pw_re,
                     type = 'link',
                     se.fit = TRUE,
                     newdata = ind_pred_data,
                     # to make specific-individual-level predictions rather than average-i
                     re.form = NULL
                     )
```

```
Warning in checkTerms(data.tmb1$terms, data.tmb0$terms): Predicting new random effect levels
Disable this warning with 'allow.new.levels=TRUE'
```

```
ave_ind <- predict(pw_re,
                   type = 'link',
                   se.fit = TRUE,
                   newdata = ind_pred_data,
                   # to EXCLUDE individual random effects from predictions:
                   re.form = ~0)
```

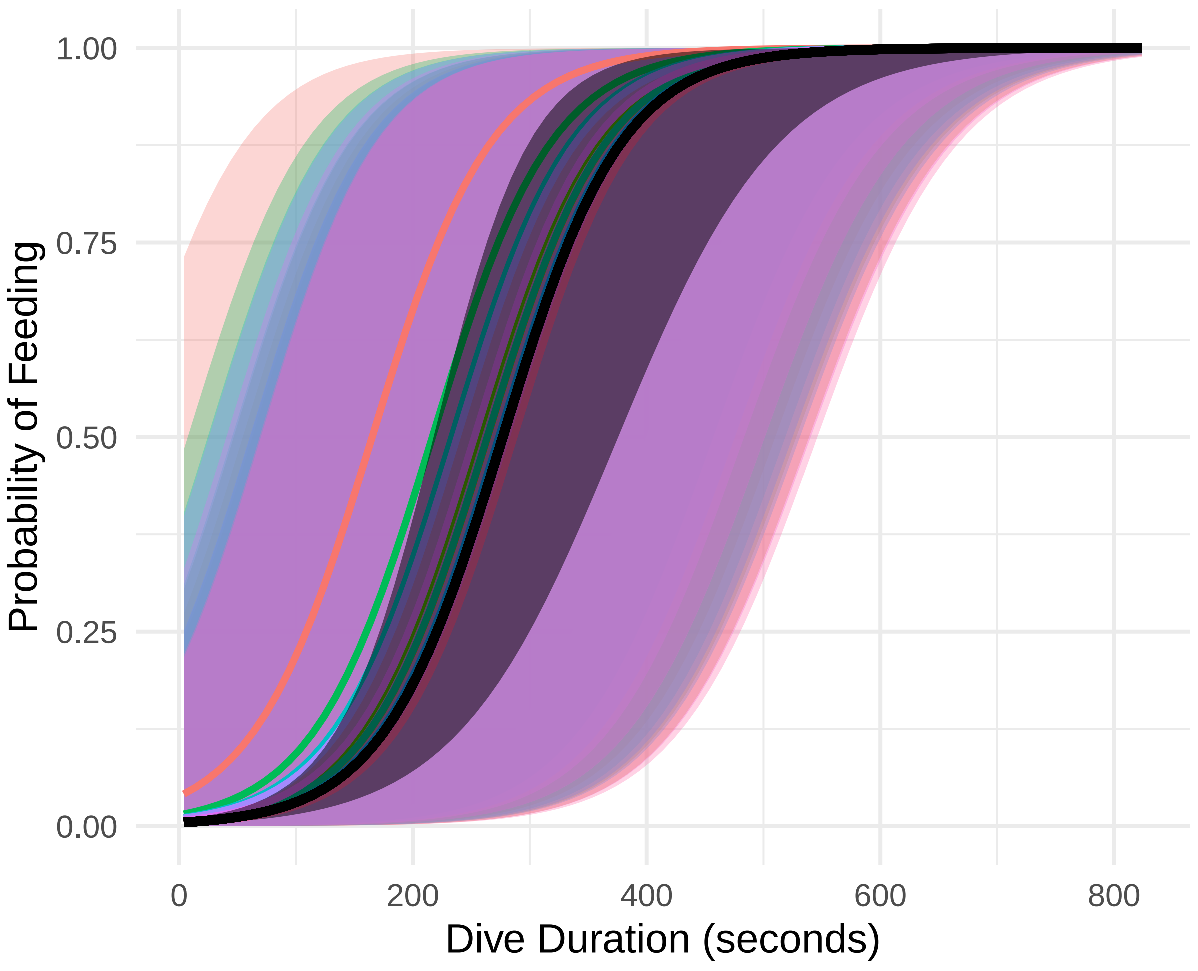
```
Warning in checkTerms(data.tmb1$terms, data.tmb0$terms): Predicting new random effect levels
Disable this warning with 'allow.new.levels=TRUE'
```

Now, as we have seen before when making prediction plots before, convert the predictions and SEs to CIs in preparation for making the prediction plot. But now, because our model has a logit link function, we need to inverse-logit transform the results with function `ilogit()` to convert from logit(p) to p – we want the predictions as probabilities at the end, not logit(probability)!

```
ind_pred_data <- ind_pred_data |>
  mutate(pred = ilogit(ind_preds$fit),
         CI_bottom = ilogit(ind_preds$fit - 1.96*ind_preds$se.fit),
         CI_top = ilogit(ind_preds$fit + 1.96*ind_preds$se.fit),
         ave_ind_pred = ilogit(ave_ind$fit),
         ave_CI_bottom = ilogit(ave_ind$fit - 1.96*ave_ind$se.fit),
         ave_CI_top = ilogit(ave_ind$fit + 1.96*ave_ind$se.fit))
```

Finally make the graph. Start with a graph like we had before (predictions for "average RE group) in black and grey), and add on a new layer colored by whale. We omit the legend because we don't really need to know which whale is which; but each color is one whale.

```r
gf_ribbon(CI_bottom + CI_top ~ duration,
            alpha = 0.3,
            fill = ~individual,
          data = ind_pred_data) |>
    gf_line(pred ~ duration,
          data = ind_pred_data,
          color = ~individual,
          size = 1.5) |>
       gf_labs(title = '',
          y = 'Probability of Feeding',
          x = 'Dive Duration (seconds)') |>
    gf_theme(legend.position = 'none')  |>
  gf_line(ave_ind_pred ~ duration,
            size = 2, alpha = 1,
            data = ind_pred_data,
            color = 'black') |>
    gf_ribbon(ave_CI_bottom + ave_CI_top ~ duration,
              fill = 'black', alpha = 0.5)
```

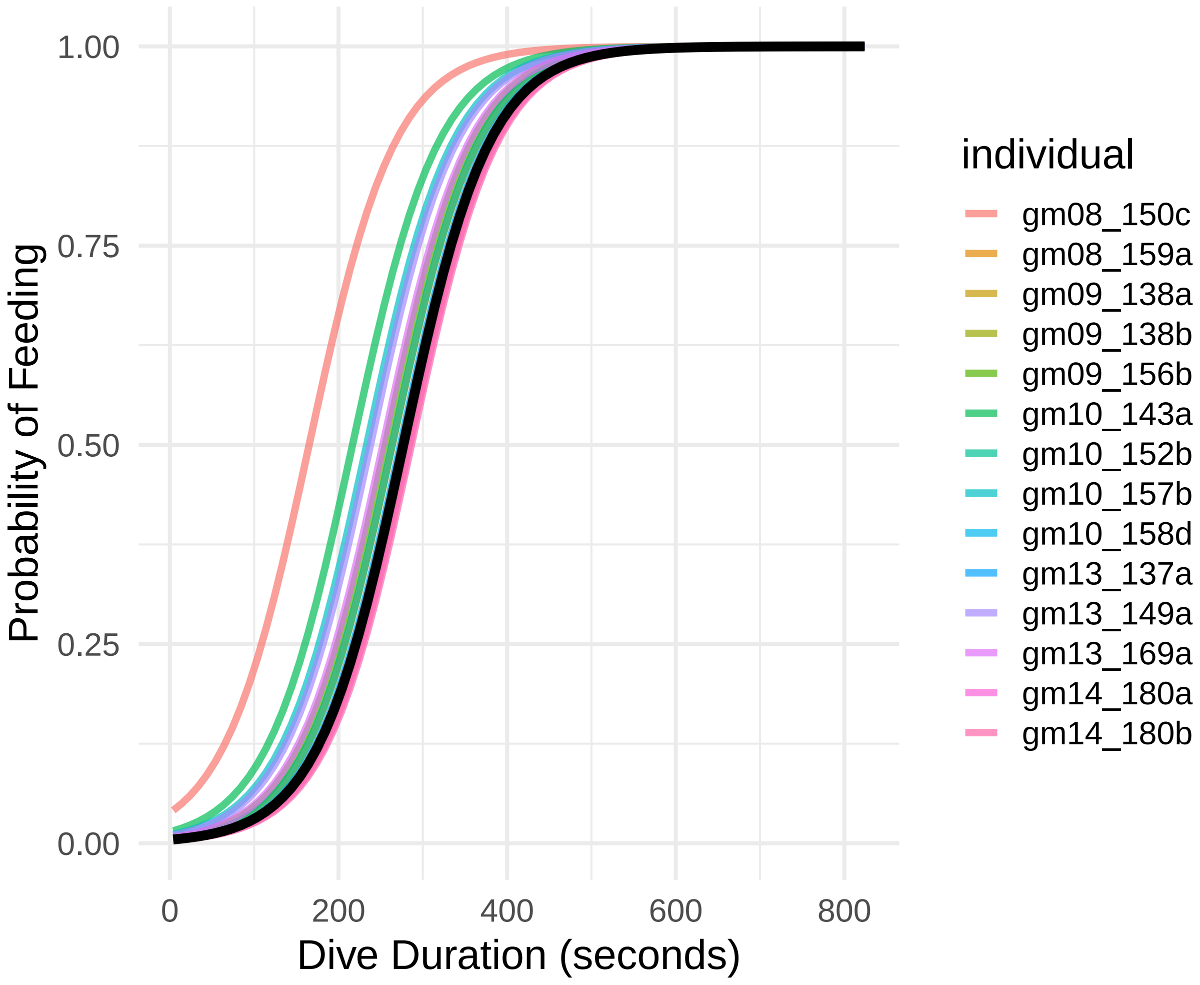


Hmm, kind of wild. There is a lot going on there and it's hard to see detail because of all the overlapping colored CIs.

What if we plot again without uncertainty, just to see the lines more clearly?

```
gf_line(pred ~ duration,
          data = ind_pred_data,
          color = ~individual,
          size = 1.5,
        alpha = 0.7) |>
      gf_labs(title = '',
          y = 'Probability of Feeding',
          x = 'Dive Duration (seconds)') |>
 #  gf_theme(legend.position = 'none')  |>
```

```
gf_line(ave_ind_pred ~ duration,
          size = 2, alpha = 1,
          data = ind_pred_data,
          color = 'black')
```



The black line is the "average RE group" and the colored ones are individual-whale average predicted probabilities of foraging.

Any ideas why the black "average RE group" = "average whale" does *not* look like it is in the middle in any sense?

Consult Prof DR with your ideas.

Then, see the very bottom of this document for my thoughts...

Prediction Plots: Population Average

So far, we have made predictions for the *average random effect group*.

But what if we wanted to make predictions for the average probability of feeding, across all whales in the population (and across all possible time-periods)? This is the kind of "average" that a model with no random effects would typically predict.

One way is to use something called a **parametric bootstrap**.

Focus on the big picture rather than the code details – we want to get *population average* predictions to compare with the average-random-effect-group ones.

To do it we will simulate many new hypothetical datasets *from the fitted model*.

To do each simulation, we draw proposed values for the intercept, all the slope coefficients, and the values of each random effect. (These will vary each simulation depending on the amount of uncertainty in the model parameter estimates (βs and the variance σ_{RE}.))

Then we re-fit the model to the new simulated data and make predictions from that model.

We repeat many times to get a sense of the *distribution* of predicted values for every desired combination of predictor variable values.

To make this work, we first have to create a function that takes a fitted model and makes the predictions we want from it.

When you run the chunk below, it won't seem to "do" anything, as what it's doing is defining a function `predict_pw_re()` for later use.

```r
predict_pw_re <- function(model){
  new_dat <- data_grid(model,
                       terms = 'duration [all]',
                       # it's important that you fill in a value *seen in the data* for ea
                       # but it does not matter which one it is
                       condition = c(individual = "gm08_150c",
                                     time_cat = "(15.2,15.3]"))
  return(predict(model,
                 newdata = new_dat,
                 type = "response",
                 allow.new.levels = TRUE))
}
```

The code below does the parametric bootstrap for you. But do not run it - it will take a half hour or so, so it's been done for you (and then the result read in from online where I stored it).

```
library(lme4)
```

```
Attaching package: 'lme4'


The following object is masked from 'package:mosaic':

    factorize
```

```
# boot_pw_re <- bootMer(pw_re, # the fitted model
#                       FUN = predict_pw_re, # our function
#                       nsim = 50, # make this 1000+ if you have time/computer can handle
#                       type = "parametric", # parametric bootstrap
#                       use.u = FALSE)
# saveRDS(boot_pw_re, '/Users/sld33/Dropbox/academic-website/static/data/boot_pw_re.RDS')

boot_pw_re <- readRDS(url('https://sldr.netlify.app/data/boot_pw_re.RDS'))
```

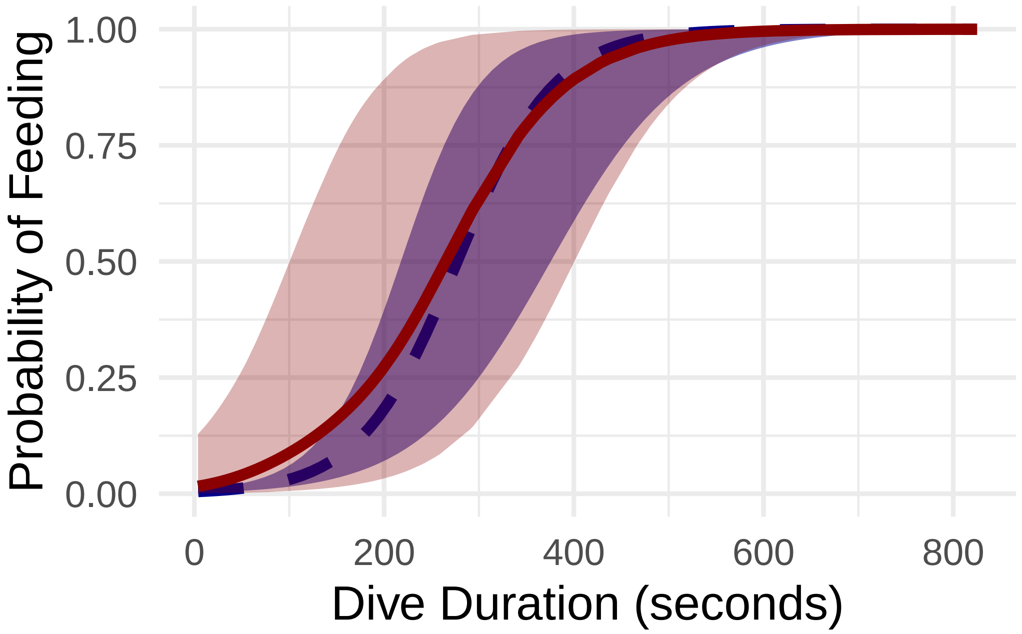
The last input to `bootMer()`, `use.u = FALSE`, tells `bootMer()` to draw new normal random effect values (simulate new whales each time, not re-use the 14 actually observed in the original dataset).

We get **a population average prediction** by taking the mean of the simulated predictions, and a CI can be estimated by taking their 2.5 and 97.5 percentiles.

```
pop_ave_data <- data_grid(pw_re,
                          # the [all] samples more values of duration
                          # to make a smoother graph in the end
                          terms = 'duration [all]')

pop_ave_data <- pop_ave_data |>
  mutate(pop_ave_pred = apply(boot_pw_re$t, 2, mean),
         CIlow = apply(boot_pw_re$t, 2, quantile, probs = 0.025),
         CIhigh = apply(boot_pw_re$t, 2, quantile, probs = 0.975)
         )
```

Now, we just want to compare the "average random effect group" predictions to the "population average" predictions:

```
gf_line(ave_ind_pred ~ duration,
        size = 2,
        data = ind_pred_data,
        color = 'darkblue',
        linetype = 'dashed') |>
    gf_ribbon(ave_CI_bottom + ave_CI_top ~ duration,
              data = ind_pred_data,
              fill = 'darkblue',
              alpha = 0.5,
              inherit = FALSE) |>
  gf_line(pop_ave_pred ~ duration,
          size = 2,
          data = pop_ave_data,
          color = 'darkred',
          inherit = FALSE) |>
  gf_ribbon(CIlow + CIhigh ~ duration,
            data = pop_ave_data,
            fill = 'darkred',
            inherit = FALSE) |>
  gf_labs(y = 'Probability of Feeding',
          x = 'Dive Duration (seconds)')
```



The population average prediction (solid red) has a somewhat different shape, and wider confidence bands compared to the blue dashed average-individual line (because it includes the individual-to-individual variation as well as uncertainty in the coefficient estimates).

Which Prediction is “Better”?

Which one is “better”? It depends.

Do you think it makes more sense to show how probability of feeding would be expected to vary over a range of depths *for a “typical” whale and a “typical” time-period* (that’s “average RE group” prediction)? Or do we want to show the feeding probability-depth relationship that we’d get if we sampled bunches of dives from many whales, at many depths (that’s “population average” prediction)?

Which does your group think is preferable in this scenario - average individual or population average? There is not a right answer. Rather, I want you to be able to understand the difference between the options, know which one you are doing, and give some kind of rationale for your choice.

Share your final conclusions with Prof DR and – you’re done!

If you have more time left, go back and repeat one or more of the prediction plots for a DIFFERENT predictor...

Why isn’t the “average” line “in the center”?

I think the “average RE group” value is far from the middle because individual whales have very different number of dives. And when NOT feeding, the whales tend to do much shorter and shallower dives, so with each row of data representing one dive, the “very shallow not feeding” behavior is more typical in terms of the proportion of dives in the dataset that have those characteristics. Also, there are a number of whales overplotted there around the “average RE group” line – the other lines that your eye is drawn to are 6-7, so there must be 7-8 more others under the black one!